JAN MCLIN CLAYBERG
PATENT AND TECHNICAL TRANSLATION

JAN McLIN CLAYBERG*
OLAF BEXHOEFT**

CERTIFIED BY AMERICAN TRANSLATORS ASSOCIATION
* GERMAN AND FRENCH TO ENGLISH
** ENGLISH TO GERMAN

5316 LITTLE FALLS ROAD
ARLINGTON, VIRGINIA 22207

TELEPHONE (703) 533-0333
FACSIMILE (703) 533-0334
JANCLAYBERG@YAHOO.COM

June 15, 2005

DECLARATION

The undersigned, Olaf Bexhoeft, hereby states that he is well acquainted with both the English and German languages and that the attached is a true translation to the best of his knowledge and ability of the German text of PCT/EP2003/004099, filed 12/11/2003, and published on 07/01/2004 under No. WO 2004/055609 A2, and of four (4) pages of amended claims.

The undersigned further declares that the above statement is true; and further, that this statement was made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or document or any patent resulting therefrom.

Olaf Bexhoeft

Olaf Bexhoeft
5316 Little Falls Rd.
Arlington, VA 22207-1522

4/pr+s
10/539885
JC06 Rec'd PCT/PTO 16 JUN 2005

WO 2004/055609
PCT/DE2003/004099

Specification

Control Method and Device and Method for Setting Up a Control System

The invention relates to a method and a device for control, as well as a method for setting up a control system in accordance with the preambles of claims 1, 2, or 29 or 36.

DE 37 07 866 A1 discloses a control device for several units of a printing press, wherein values measured at the units are supplied to a computer unit, are stored in a memory and made visible at a touch display screen. The direct control of the units is made possible by means of the touch screen and the computer unit.

A digital system for regulating and controlling a power converter and drive mechanisms is known from the publication "Siemens Energie & Automation 8 (1986) Heft 2", pages 119 and 120, wherein a regulating and control device is designed by means of design software, instead of being programmed. The designed regulation and control functions of this one regulating device are translated into a list code, the application program is processed together with the system software by the regulating and control device. The regulating device is designed in a structural image-oriented design language of required hardware and software components (page 121 to 123).

A system for the rapid digital regulation and control of power converter drive mechanisms is disclosed in the above mentioned publication on pages 112 to 115, wherein the hardware and software is constructed in modular structures (pages 116 to 118).

DE 197 40 974 A1 discloses a book production system with a program having an object structure and a memory, in which inter alia information regarding the machinery contained in the installation, as well as setting parameters, are stored. To manufacture a defined product, the setting parameters destined for this product are output to the control of the machine by the program by means of the data stored in the data bank.

A system for transmitting OPC data via the internet to an OPC server of an automating system is known from DE 100 36 552 A1.

An article by Wolfgang Weber "Verteilte Systeme - Verteilte Objekte - DCOM" [Distributed Systems - Distributed Objects - DCOM] at a seminar on data processing WS 1989/99 of the Ruhr University at Bochum, Chair for Data Processing, deals with the employment of DCOM running time systems in connection with the communication between several different computers and processes.

In the article "Useware in der Praxis: Die DICOweb Bedienung - ein Beispiel für eine ergonomische Gestaltung in der Drucktechnik" [Useware Actually Employed: DICOweb Operation - an Example of an Ergonomic Design in Printing Technology] by Gregor Enke, 44 (2002), an operating concept for a printing press is introduced. The concept provides a manipulation-oriented operation in the course of the product preparation and the influencing of process values.

The object of the invention is based on creating a method and a control system, as well as a method for setting up a control system.

In accordance with the invention, this object is attained by means of the characteristics of claims 1, 2, or

29 or 36.

The advantages to be gained by means of the invention consist in particular in that for one the architecture of the control system is constructed decentralized in such a way that it supports a modular and flexible construction or design of the processing machine.

Simultaneously there is a central data management device whose structure is adapted from the existing configuration of the processing machine, or reflects it, which makes possible a simple centralized configuration across distributed systems.

The central data management system and/or the architecture of the control system form a scaleable system. In this way the individual steps for processing the data can be scaled on a single computer or across several computers depending on the application case and/or workload. Therefore the system, in particular the software and the hardware, covers a large spectrum of different types and sizes of machines (for example printing presses for newspapers, jobbing, sheets, securities etc.). The software, or the data inventory of the control system, is configured as a function of the type of application, for example from predefined modules and data inventories, in particular designed. Separate programming of data of each individual configuration can be omitted, only the design of the control system with previously known components to correspond with the installation takes place. The modular architecture and the type of planning and implementation can be advantageously expanded to installations with several sections.

Regarding the variability and flexible application for the most diverse machine types, the standardized, almost

automated design of the control system, in particular the data memory, is of advantage. The program portion of the data server, which is identical to a large extent, except in regard to different types of machines, forms the machine-specific structure in the data server actually on the basis of a data set describing the installation, i.e. the data server only receives its identity (for example through its memory element which was previously "empty") in respect to the installation to be controlled, and is set up accordingly. This set-up of the base structure can be followed during the later operation in the course of operating the installation by a change of applied parameter specifications and/or an activation or deactivation of basically provided components, or units. In what follows, a differentiation will be made between the implementation (i.e. data or software configuration) of the machine-specific data structure into the control system, and an adaptation, to be made during the running of the installation, to the product to be manufactured, which are in part also called "configuring" or "setting-up" in the prior art. For example, control console computers, or their memories, have a fixedly programmed identifier space when the installation is being delivered, for example, while with the present system there is a freely programmable identifier space, whose data structure - for example by implementing the config files - is only then set up. For this purpose, the data server has a process on the one side, and on the other side the config file characterizing the installation.

In an advantageous embodiment of the control console, it has an operating console which makes it possible for the operator to recognize the colors of the actual side and to

change them. In further development, the values for regulating and controlling the machine can be directly changed or input by means of a touch screen-capable display field at the display screen.

Exemplary embodiments of the invention are represented in the drawings and will be described in greater detail in what follows.

Shown are in:

Fig. 1, a schematic installation with a control device,

Fig. 2, a schematic representation of the design of an installation,

Fig. 3, a schematic representation of the architecture of the control device,

Fig. 4, a schematic representation of the architecture of a multi-section installation.

An installation 01, for example a printing press 01 or printing press installation 01 as represented in Fig. 1, has a number of units 02, 03, 04, for example installation elements for performing defined method steps, such as a material supply device 02 (for example a roll changer 02, or several roll changers 02.1, 02.2, 02.3, etc.), printing units of printing towers 03, units 04 for further processing, such as folding apparatus 04, etc. In turn, the printing units 03 can have several lower-order units 05, for example printing groups 05, for example identified by 05.1, 05.2, 05.3, 05.4, etc. The different units 02, 03, 04, 05 can each be provided once or several times in the installation. The number and/or embodiment of the units 02, 03, 04, 05 can be different for different embodiments of the installation.

The installation 01 furthermore has a symbolically represented control system 06, which controls the units 02,

03, 04, 05, or their drive mechanisms, settings, etc. individually and in the interaction between each other. Inter alia, the symbolically represented control system 06 contains, as will be explained in detail below, various elements, such as one or several memories, one or several computing units, as well as signal connections between the elements, such as data busses and/or protocol converters, for example. The control system 06 is in a signal connection, for example via connections 07, with the units 02, 03, 04, 05 or, if provided, with controls 08 and/or regulating elements 08 assigned to them.

Depending on the type, design or application, such an installation 01 is differently configured regarding its units 02, 03, 04, 05, the respective number of units and/or the detailed specification of the hardware of the units themselves. This means that this relates, for example, to a jobbing printing press with a defined number of roll changers (of a defined type), a defined number of printing groups in a horizontal arrangement (of a defined type), a folding apparatus (of a defined type), etc. In connection with the hardware it is intended here to understand that the configuration of the installation 01 is the actual design of the installation with actually provided units 02, 03, 04, 05, and possibly their design.

Information regarding such a configuration of the entire actually existing installation 01 is centrally stored with the corresponding data in the control system 06. The data are preferably centrally stored in a centralized data management device 09, for example a memory 09 or data server 09. In an advantageous embodiment, these data (for configuring the installation) can be implemented and/or

changed via at least one input interface 11. In an advantageous embodiment, the data for configuring the installation are provided as a data set F (for example file F, in particular configuration file F) by means of a data processing unit 16, which is independent of the control system 06, for example a computer 16 and, following their completion, are transmitted via the input interface 11 to the central data management device 09, or are implemented there. The preparation of the configuration file F is used for the mapping by means of control technology of the actual installation configuration and will be understood in what follows as "projecting" the control system 06 in view of the actually existing installation 01, or its units 02, 03, 04, 05. For example, the transmission of the file F can also take place via a network by the manufacturer, for example via the internet. However, the control system 06 can also have its own means for projecting the control system 06, or for the preparation of the data set F which, for example, substantially correspond to those of the data bank and/or surface explained in greater detail below.

The mapping of the configuration in a configuration file F, i.e. the projection of the control, takes place in an advantageous embodiment by using a memory unit 17, in particular a data bank 17, with a stock of pre-known or predefined objects 12, 13, 14, which for example are assigned to printing press types and/or embodiments of the above mentioned units 02, 03, 04, 05. The installation 01, for example a printing press 01, is then projected in such a way that it is put together from the stock of predefined objects 12, 13, 14 (for example printing towers, roll changers, folding apparatus, etc.), wherein the data (at least basic

data), properties and/or sub-programs required for later operation of the installation 01 are assigned to these objects 12, 13, 14, or are allotted them by selection. For example, these specific data and sub-programs are present in the data bank 17 and are linked to the respective object 12, 13, 14. More complex objects 12, 13, 14 at least contain the possibility of further details, in which lower-order objects 13.1, 13.2, for example specific variants, specific embodiments, details, specific settings, etc., here for the example of a "printing unit1", "printing group1" and printing group2", on a lower level are assigned to the objects 12, 13, 14, for example in the manner of a tree structure or register structure.

Further details can exist on more than one lower level in the tree structure, as represented in Fig. 2 for the example of the printing unit 03 with "lateral registration" and "circumferential registration", as 13.2.1, 13.2.2. A further, not further identified level represents, for example, "parameters", with still lower located "parameter1" and "parameter2". Basic data and/or sub-programs are assigned to all these specifications in the data bank 17 which, when selecting the appropriate object 12, 13, 14 (with sub-specification, lower-order objects, etc.) are transferred to a data set F. Data, or finally selected parameters, can be differently selected for different units 02, 03, 04 of the same type, or can be predetermined by being entered. As represented by way of example in Fig. 2, the installation 01, which is to be projected (by means of software technology), has three roll changers 02, two printing units 03, as well as a folding apparatus 04, wherein the parameters from the object stock 19, schematically represented on the right side

of the figure, are assigned to each unit 02, 03, 04, 05. Following the selection of the components/objects intended for the installation 01, all assigned data (parameters)/sub-programs, are transferred in accordance with a predefined pattern to a data set F or a file F (for example a "config file" F). This data set F then represents a map of the installation 01 which is actually to be operated and then has, for example, all preset values, preset command variables, etc. essential for the operation and control and can be available in text or in binary format.

In an advantageous embodiment, the projection is performed on the data processing unit 16 which is independent of the control system 06 or can be separated from it, for example on a computer 16. In this case an embodiment is of advantage wherein a program surface 18 has a linkage with the above mentioned data bank 17 in such a way that a selection or copy of an object 12, 13, 14 (or its identification or name) also contains the specific data or parameters and/or sub-programs and/or sub-objects 13.1, 13.2 (furthermore also process variables 12, 13, 14, 13.1, 13.2, 13.2.2, 13.2.2) - comparable to the embedding of an OLE object. An embodiment of the program surface is particularly easy to use, by means of which objects 12, 13, 14 offered in an object stock 19 in a tree structure can be further differentiated by selection, and the objects 12, 13, 14, or the detailed objects 13.1, 13.2 can be supplied by copying, in particular "drag and drop", to the installation to be projected in the form of its own display screen area 21. In the background of this operation, or at the end, the data and procedures assigned to these objects 12, 13, 14 with lower-order objects 13.1, 13.2, etc. are copied - for example in a predeterminable standard

format - to the data set F. The file F created on the basis of the selected process variables 12, 13, 14, 13.1, 13.2, 13.2.2, 13.2.2 can now be fed after its completion to the control system 06 of the installation 01, in particular to the data server 09. It contains, for example, application names predetermined by the user - and preferably standardized - as variables.

For one, a data processing operation (or data processing routine) has been advantageously implemented in the data memory 09, which is designed to produce a data structure by means of projection data which, for another, are available to it. In the present case, the configuration file F which had been transferred to it, is available to the data memory 09, by means of which a software-related configuration (establishment) of the memory takes place which is custom-made for the installation 01 mapped by the file F. This means that following the transmission of the configuration file F into the data memory 09, first a data structure is created, which is matched to the installation 01, instead of merely filling an already fixedly programmed memory environment of open parameters with data. Such a fixedly programmed memory environment would have to be individually programmed to a large extent for each machine type and/or every more extensively differing configuration of the hardware. By means of the advantageous design of the data memory 09 with a data processing operation and configuration file F assigned to each other, the custom-made design of the data structure, and in the end of the control system 06, is possible in a simple way. The actual installation 01 is mapped in accordance with fixedly predetermined routines of the data processing operation by means of the file F (process

variables 12, 13, 14, 13.1, 13.2, 13.2.2, 13.2.2) in the data memory 09 or the control system 06. Only then is the data and program structure formed as a system of the data memory 09 as such.

The advantage of this double solution is that, on the one hand, procedures, actions, behavior and/or logical connections have been set up for the software portion and higher-order control processes and, on the other hand, the configuration data of the file F define the components and sub-routines of the actually existing installation 01. In this way changes or error corrections of the "hardware" can be achieved solely by implementing a new file F, and improvements or error corrections of the data processing operation solely by implementing a portion of, or the entire data processing operation. For example, in case of an additional or replacement unit of a new type it is only necessary to read in a new config file F for the new installation 01, taking this change into consideration, the program portion of the data server 09 recognizes the mapped installation 01 and adjusts its memory area and, if required, other components of the control system 06, accordingly. In this case no change in the program source codes in respect to parametrization, routines, etc., needs to be made. The program portion (data processing operation) is designed in such a way that the data structure and the functionality of the installation 01 regarding different machine types is formed solely by means of the available config file F.

The described projection goes far beyond the conventional programming of a control console computer or section computer wherein, although the installation 01 is also mapped by software, the routines and the program source

codes containing the parametrization are, tightly matched to the machine, fixedly programmed in the respective computer unit. The installation 01 can then be operated via an operator interface, this means that customarily already entered components can be switched on and off by means of software, parameters can be changed, etc. In the prior art this is also sometimes called "configuring the installation 01", but means the selection of components and/or parameters already fixedly set in the program, and not the "configuration by means of software" of the data server 09 or the control system 06 in the manner called "projecting" here. In contrast to the so-called "configuration of the installation 01", wherein the operating data in a fixedly preset data structure are selected and are thereby "configured", with the present way of operating the data structure of the data server 09 itself is configured, i.e. the data server 09 itself is designed so it can be configured (projected). Projection is to be understood as the mapping of the actually existing units - and only these - in an appropriate data structure.

In an advantageous embodiment it has of course been provided that, following the projection and implementation of the file F, the operators can switch units implemented through an interface 15 off or on, or can change parameter values via a control level 41 (see below) and/or a partially or fully automated system (such as a product planning system), i.e. can perform a so-called "configuration of the installation 01" in the above mentioned way from the units now entered in the data server 09.

In an advantageous further development, in the course of projecting the installation 01 communication-specific

information (for example interface protocols used) regarding the units 02, 03, 04, 05, and if required regarding the hardware components of the control system 06 used, are read out of the object stock. In a first variation, these data can be integrated in the config file F, can also be implemented in the data server 09 and fed from there to a communication server 23, explained in greater detail below, for its "configuration by means of software". In an embodiment, indicated by dashed lines in Figs. 2 and 3, for the configuration by means of software of the communication server 23 in the course of the projection, at least one second file F', for example configuration file F' - config file F' for short, is established, which is either indirectly implemented in the communication server 23 via the data server 09, or via an interface, not represented, of the communication server 23. Communication-specific information regarding the projected units 02, 03, 04, 05 is then available to the latter.

A systematy leading away from a one-time production is created by the procedure and embodiment for setting up the control system 06, which can be used in connection with the most diverse products of a manufacturer of the installation 01, for example different types, product lines and design or equipment stages, in a manner which is simple and low in errors. The person projecting the control system 06 by means of the planned installation 01 no longer needs to perform detailed programming, custom-tailored to the installation 01, but instead only maps the components or units 02, 03, 04, 05 of the installation 01. Programming takes place by taking in the data regarding the object stock 19, the implementation in the data server 09, as well as the program portion of the

data server 09 which is substantially constant regarding the different types, product lines and design or equipment stages.

The data memory 09 is advantageously designed as a data server 09 with an open interface, in particular as a data server 09 (OPC data server) with at least one open OPC interface 15 for data exchange on the basis of OLE/COM and DCOM. The data memory 09 manages the objects or process variables 12, 13, 14, 13.1, 13.2, 13.2.2, 13.2.2 on the basis of an object management, for example in accordance with the COM (component object model)/DCOM (distributed component object model) standard. This means that it has an operating system which supports a method for inter-process communication (object request mediator) which is designed to exchange complex data structures. In the example, the operating system is provided in the form of Windows NT4.0(R) (or higher), or Windows 2000(R) (or higher) and as standard for COM/DCOM communication. However, it can also be a comparable combination of operating system and object request mediator meeting the above mentioned conditions, for example the operating system LINUX(R) and the standard CORBA(R). This equally applies to the element described in what follows, for which the use of Windows 2000(R) together with COM/DCOM has been called advantageous or a requirement.

The data can be exchanged via the open interface 15 in accordance with the data exchange method OLE (object linking and embedding), for example with a network, field bus, an application and/or a visualization. The at least one open OPC interface 15 allows access by means of a further external unit 20, module 20 or application 20 (for example of a dryer, a print pre-stage and/or a roll supply), called "consumer" in

what follows, in that it is brought into contact, is connected with the data server 09. The additional consumer "helps itself" autonomously to the data from the data server 09.

The objects, or process variables 12, 13, 14, 13.1, 13.2, 13.2.2, 13.2.2 are mapped by the data server 09 in an identifier space and managed. The administration can consist, for example, of memorizing, archiving and reconstructing process data and process variables 12, 13, 14, 13.1, 13.2, 13.2.2, 13.2.2. The identifier space is now specifically configured to correspond to the projected installation 01, particularly again in a tree structure. The data server 09 can additionally be designed so that errors detected in the control and/or regulation of the installation 01 are entered in a data set, for example a so-called log file.

In an advantageous embodiment the data server 09 is integrated into an architecture for the control system 06 explained in what follows (Fig. 3).

The data server 09 is connected via a further interface and a signal connection 22, for example a network 22, with at least one process or computing unit 23, for example a server 23, in particular a communication server 23. In regard to the control system 06 with decentralized components (for example lower-order processes 24 and/or controls 08, see below), the process or computing unit 23 represents a higher-order process or computing unit 23. In it, a conversion of the so-called raw data made available by the data server 09 into the communications protocol respectively demanded for the lower-order processes 24 and/or controls 08 takes place in it in a higher-order process. The network 22 between the

data server 09 and the higher-order process or computing unit 23 is for example designed as an ethernet with a transmission rate of at least 10 Mbit/s, for example. Unless stated differently, a "network" is understood to be a closed network in accordance with hardware technology of a uniform net type. In an advantageous embodiment a stochastic access method, in particular the CSMA/CD access method standardized in accordance with IEEE 802.3, is used as the access method. In principle, the communication can be based on diverse protocols, but in an advantageous embodiment on the TCP/IP protocol or a socket connection.

The communication server 23 is generally understood to be a process unit 23, which is a mediating layer between the server 23 and network-specific processes located "below" it. For example, the server 23 here functions as client (OLE or COM/DCOM client), which can pick up objects or process variables 12, 13, 14, 13.1, 13.2, 13.2.2, 13.2.2 from the data server 09 (OLE server). It receives and/or processes the objects or process variables 12, 13, 14, 13.1, 13.2, 13.2.2, 13.2.2 also on the basis of an object management in accordance with the COM/DCOM standard and is machine-specifically configured. The communication server receives the machine-specific configuration for example in the above mentioned way through the projected config file F'.

Thus, the server 23 forms a "communication layer" between the data server 09 and clients located "below it". These can be, for example, one or several controls 08 combined into one or several networks 28 and are connected, for example, via appropriate interfaces 27 or nodes 27 with the respective network 28 or signal connection 28. The server 23 constructs for example a direct signal connection

(not represented) with these controls 08 used as communication partners, or maintains it. On the one hand, it interprets data or work order ("jobs") received from the control 08 and transfers these data to the data server 09. On the other hand, it converts data to be transmitted from the data server 09 into jobs and transmits these to the respective control 08. The controls 08 can be designed as programs running on a PC, as SPS units, or in another way. The job receives from the server 23 the information regarding the node 27 affected by the job, for example in the head of the respective network protocol.

The signal connections 28 are advantageously designed as one or more networks 28. They can lead in a star shape (as represented in star topology) to the controls 08, or (not represented) each can serve several controls 08 in a bus or ring structure.

The network 28 is embodied in an advantageous manner as a network 28 with a deterministic access method, in particular based on token-passing, for example as an arc net 28. In this case the server is embodied for converting the jobs into the protocol used. If the network 28 is embodied as an arc net 28, the communication server 23 is designed as an arc net server. In an advantageous variation, the connection 28 can also be designed as a so-called "profi bus system".

In the exemplary embodiment of Fig. 3, the server 23 is not in a direct signal connection with the controls 08, but instead in an advantageous manner via several lower-order processes 24 or data processing and/or computer units 24, three in this case. The processes 24 (or also process units 24) can be housed, as represented in separate units 24.1,

24.2, 24.3, 24.4 (i.e. in their own processors, in particular communication processors, or even in their own housings or computers), or in a common component, for example a computer. The processes 24 are embodied to serve a network of a defined type and constitute so-called "net handlers" 24. The net handlers 24 are interchangeable and can be embodied depending on the net type to be served (for example profi bus, interbus-S or real time ethernet). Net handlers 24 of different types can also be connected at the same time with the server 23. For example, in Fig. 3 the net handlers 24.1, 24.2, 24.3 can be designed as arc net handlers, while an additional net handler 24.4 supports another net type and/or another protocol, for example, in order to communicate with units, or their controls 08, on the basis of this net type and/or protocol.

A connection 29 between the server 23 and the lower-order data processing and/or computing units 24 is designed, for example, as a network 29, here in star topology. The information as to which node 27 is in a signal connection 29 with which lower-order data processing and/or computing unit 24 (processes 24), for example, has been implemented in the server 23. The information is then sent, for example in the manner of a "switch", to the lower-order process 24 concerned. In a variation, all information is sent to all lower-order processes 24, wherein acceptance and further processing is decided, for example, by means of the information in the protocol head. The information regarding which control 08 is to be addressed over which lower-order data processing and/or computing unit 24 (process 24), can be implemented in the lower-order control 08 itself and can be changed.

In the example, the network 28 is embodied between the lower-order data processing and/or computing unit 24 and the connected control 08 as a network 28 in star topology. The lower-order data processing and/or computing unit 24 is designed, for example, as server 24 with COM/DCOM object management. In further development, each server 24 contains a program 31 designed as a driver 31, or driver software 31, which supports the operating system NT4.0(R) (or higher), or Windows 2000(R) (or higher). In this way object-oriented processing with COM/DCOM object management is made possible on all levels (corresponding to the above mentioned LINUX/CORBA or comparable system), A switch 33, a so-called switch 33 or switching hub, can be arranged downstream of the data processing and/or computing unit 24, which evaluates the target address of the data packet (here the control 08 or the node 27 concerned) and passes the data packet on specifically only to this control.

In addition to the general connection between the data server 09 and the individual controls 08, logical connections 32 between the controls 08 with "short" logical paths are provided (cross communication). These are used for communication during the control process, in which no jobs are needed from the data server 09.

The represented architecture makes it now possible to configure the installation 01 in a simple manner during planning, or to project the control system 06 corresponding to the installation 01, and to implement the resulting settings and data via the data server 09 for the operation. The embodiment with several lower-order data processing and/or computing units 24 makes it possible to couple the different controls 08 freely and arbitrarily to one of these

data processing and/or computing units 24 (and to store this information there, if required). Depending on the load, possibly in connection with interferences or in the course of expanding the installation 01, the controls 08 can be assigned to the different, or to one data processing and/or computing unit 24 to be expanded, or to a further process 24. Therefore the concept can be freely upwardly scaled. If the installation 01 is in the planning stage, it is already possible in the course of projecting to decide on the number of the lower-order data processing and/or computing units 24 (net handlers 24, for example as arc net handlers), as well as the intended assignment of the controls 08, and to already take them into consideration in the file F (or in a separate configuration file for the data processing and/or computing units 24). A presetting is then already implemented in the server 23 and/or the net handlers 24, which is simultaneously taken into consideration in the network plans for the connections. It is of particular advantage that, independently of the multitude of units 02, 03, 04, 05, all basic settings, actual values needed for control and new command variables of a section are implemented in the common server 23, or changed data are stored there. The control level 41, as well as the lower-order systems, can always access this data stock via the represented architecture.

In a further development, as represented in Fig. 4, the mentioned concept can be expanded to an installation 01 with several sections 34, i.e. basically machines 34 which can be operated independently of each other. In this application the installation 01 has several data servers 09, for example one for each machine 34 or section 34. The data servers 09 are each connected via network(s) 29 (ethernet in the

example) with a communication server 23. Further lower-order data processing and/or computing units 24 (net handlers 24), not represented here, can be assigned to each communication server 23. The controls 08 are again in signal connection with the communication server 23 (directly or via a lower-order net handler 24) via a network 28. In the example a second network 36 per machine 34 is provided, into which further control- or operation-relevant devices 37, 38, 39 can be integrated.

The data processing units 23 (here arc net servers 23) are in signal connection 32 with each other for communication. It has furthermore been provided that a section-overlapping information exchange between the various data servers 09 and the communication servers 23 takes place on a control level 41 of the installation. As in the example in accordance with Fig. 3 it is furthermore provided that a cross-communication 32 between the controls 08 within a section 34 and/or in a section-overlapping manner, can take place on the level between the communication servers 23 and the controls 08.

The control level 41 of the installation 01 is symbolically (in dashed lines) indicated in Fig. 3 above the control system 06. The components of the control level 41 are a control console 42, 43, 44, which, for example, has at least a computing or data processing unit 42 (control console computer 42), a visualization device 43 (display screen 43), as well as an operating console 44. The control console 42, 43, 44 is used for the communication of the operators with the installation 01, or with a section 34 of the installation 01. The control console 42, 43, 44, or portions of the control console 42, 43, 44, are in signal connection with the

data server 23, for example via an open OPC interface in accordance with the above explanations.

List of Reference Symbols

01	Installation, printing press, printing press installation
02	Unit, material feeding device, roll changer
02.1	Roll changer
02.2	Roll changer
02.3	Roll changer
03	Unit, printing unit, print tower
04	Unit for further processing, folding apparatus
05	Unit, printing group
05.1	Printing group
05.2	Printing group
05.3	Printing group
05.4	Printing group
06	Control system
07	Connection
08	Control regulation, communication partner
09	Centralized data management, data memory, data server
10	-
11	Input interface
12	Object, process variable
13	Object, process variable
13.1	Lower-order object, process variable
13.2	Lower-order object, process variable
13.2.1	Further detail, process variable
13.2.2	Further detail, process variable
14	Object, process variable

15	Interface, open, OPC interface
16	Data processing unit, computer
17	Data bank
18	Program surface
19	Object stock
20	Unit, module, application
21	Display screen area
22	Signal connection, network
23	Process or computing unit, server, communication server, process unit
24	Lower-order process, data processing and/or computing unit, net handler, server, process unit
24.1	Lower-order process, data processing and/or computing unit, net handler, server
24.2	Lower-order process, data processing and/or computing unit, net handler, server
24.3	Lower-order process, data processing and/or computing unit, net handler, server
24.4	Lower-order process, data processing and/or computing unit, net handler, server
25	-
26	-
27	Interface, node
28	Signal connection, network, arc net
29	Signal connection, network
30	-
31	Driver, program, driver software
32	Logical connection, cross communication
33	Switch
34	Section, machine

35	-
36	Network
37	Control-/operating-relevant device
38	Control-/operating-relevant device
39	Control-/operating-relevant device
40	-
41	Control level
42	Computing and data processing unit, control console computer
43	Visualization, display screen
44	Operating console
F	Data set, file, config file.